



Fall 2014 SEI Research Review

Malware Analysis

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Jonathan Spring
October 29, 2014



Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 29 OCT 2014		2. REPORT TYPE N/A		3. DATES COVERED -	
4. TITLE AND SUBTITLE Fall 2014 SEI Research Review: Malware Analysis				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Jonathan Spring				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited					
13. SUPPLEMENTARY NOTES The original document contains color images.					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 13	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Copyright 2014 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon®, CERT® and CERT Coordination Center® are registered marks of Carnegie Mellon University.

DM-0001729



Investigators

Principal Investigator

Ed Stoner
CERT Division
ers@cert.org
(412) 268-6187

SEI Investigators

William Casey, PhD
CERT Division
Sagar Chaki, PhD
Software Solutions Division
Cory Cohen
CERT Division
Jeffrey Gennari
CERT Division

Arie Gurfinkel, PhD
Software Solutions Division
Jeff Havrilla
CERT Division
Charles Hines
CERT Division
Leigh Metcalf, PhD
CERT Division
Soumyo Moitra, PhD
CERT Division
Jonathan Spring
CERT Division
Rhannon Weaver, PhD
CERT Division



The Problem

There is a lot of malicious software

- Hundreds of thousands of new, unique samples collected globally

But malware analysis is a time-consuming process

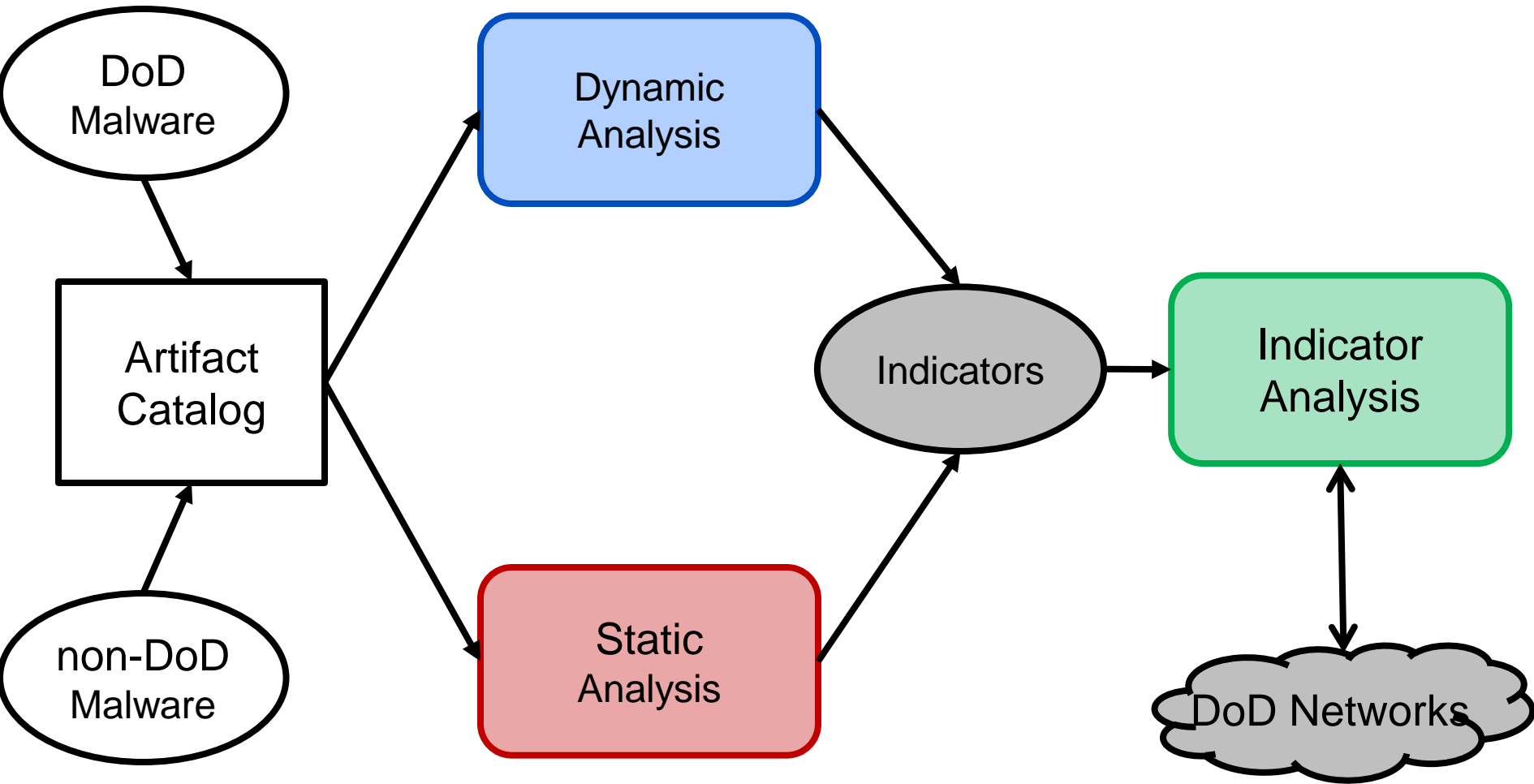
- And human-intensive

So we need better automation to understand the threat.

- Automated static analysis of artifacts
- Large-scale analysis of indicators



Malware Analysis Process



Static Analysis Improvements

1. Compiler transformation framework

- ROSE [Quinlan 2000]
- Well-established program analysis technique

Implemented to analyze malware binaries at a larger scale

2. Optimize suffix-tree data structures for the identification of longest common substring (LCS)

We do substring searches a lot, and it takes a long time

Helps with:

- Malicious code analysis (code-clones)
- Zero-suppressed binary decision diagrams (ZDDs) for compact representations of set families.



Dynamic Analysis Improvements

Malicious Behavior and Model Checking: Describe formally software behavior and be able to determine if the behavior is malicious.

1. Construct an accurate binary instrument for trace capture (trace monitor)
2. Use trace monitor to capture benign and malicious software behavior (collect trace data)
3. Analyze trace data to determine features that link software by behavior.
4. Formally model methods to classify software traces as malicious or benign within the formal language of hyperproperties.



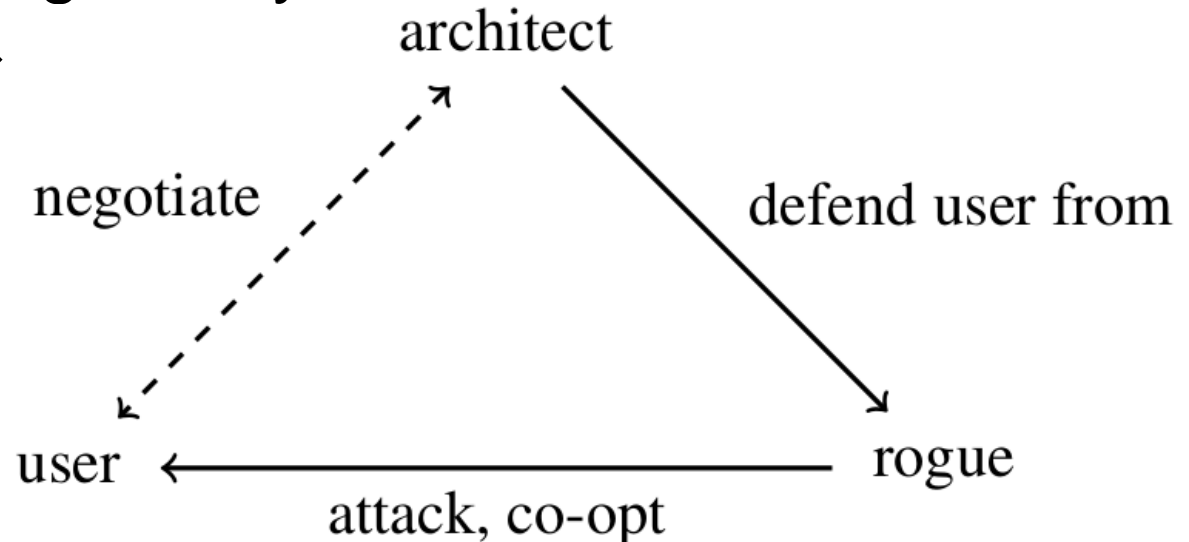
Indicator Analysis Improvements

Lead by doing Discovery at Scale

- Passive detection of Misbehaving Name Servers
- Route Injections – What are they good for?
- Everything You Wanted to Know About Blacklists but Were Afraid to Ask

Lead by codifying theory and models

- Game theory →
- Metrics
- Take-down models



Global Improvements

How do we analyze and design observations of engineered artifacts?

Usually, a scientist would turn to philosophy of science to answer methodological questions

But there were no answers in the philosophy literature

- Thus our paper "Exploring a Mechanistic Approach to Experimentation in Computing."

Computing is new and old

- Newer – study of engineered mechanisms
- Old – study of physical mechanisms

Accommodating these differences presents fundamental challenges we are just unravelling.



Results

Jin, W., Chaki, S., Cohen, C., Gennari, J., Gurfinkel, A., Havrilla, J., Hines, C., Narasimhan, P.: [Recovering C++ Objects From Binaries Using Inter-Procedural Data-Flow Analysis](#). 3rd ACM SIGPLAN Program Protection and Reverse Engineering Workshop (PPREW 2014). 2014.

Casey, W., Morales, J., Nguyen, T., Spring, J., Weaver, R., Wright, E., Mishra, B. ["Cyber Security via Signaling Games: Toward a Science of Cyber Security,"](#) 10th International Conference on Distributed Computing and Internet Technology, Bhubaneswar, Odisha, India, February, 2014.

Casey, W., Wright, E., Morales, J., Appel, M., Gennari, J., Mishra, B. ["Agent-based Trace Learning in a Recommendation-Verification System for Cybersecurity"](#) IEEE International Conference on Malicious and Unwanted Software (MALCON) 2014, Fajardo, Puerto Rico, Oct 28-30, 2014.

Hatleback, E., Spring, J. "Exploring a Mechanistic Approach to Experimentation in Computing." Philosophy & Technology. Springer. 2014. DOI 10.1007/s13347-014-0164-9.

Spring, J. "Toward Realistic Modeling Criteria of Games in Internet Security." Journal of Cyber Security & Information Systems. Vol 2, num 2. CSIAC. 2014.

Cohen, C. Practical Problems in Automated Static Analysis of Malware, Dagstuhl Seminar #14241, June 10-13, 2014.



Future

This line-funded work was not renewed per se

The work will be continued as:

- Customer-funded deliverables
- New directions within LENS work



Contact Information

Ed Stoner

Senior Member of the Technical Staff

CERT/CC – Threat Analysis

Telephone: +1 (412) 268-6187

Email: ers@cert.org

U.S. Mail

Software Engineering Institute

Customer Relations

4500 Fifth Avenue

Pittsburgh, PA 15213-2612

USA

Web

www.sei.cmu.edu

www.sei.cmu.edu/contact.cfm

Customer Relations

Email: info@sei.cmu.edu

Telephone: +1 412-268-5800

SEI Phone: +1 412-268-5800

SEI Fax: +1 412-268-6257



References

Quinlan, D. “ROSE: A Preprocessor Generation Tool for Leveraging the Semantics of Parallel Object-Oriented Frameworks to Drive Optimizations via Source Code Transformations,” 383-397. *Proc. Eighth Int’l Workshop on Compilers for Parallel Computers (CPC ‘00)*. Aussois, France, Jan. 2000, École Normale Supérieure, 2000.

